

REMARKS/ARGUMENTS

The claims have been amended as set forth above. Applicants believe that the amendments to the claims include allowable subject matter. Applicants respectfully request reconsideration of the claims. Applicants reserve the right to pursue the subject matter in any of the cancelled claims in any forthcoming continuation application(s).

I. Examiner Interview Dated October 7, 2008

An interview was held on October 7, 2008. During the interview, the current changes were discussed. Applicants believe that an agreement was reached that the current changes overcome the rejections set forth herein.

II. Rejection Under 35 U.S.C. § 103

Claims 1, 6-8, 10, 12, 14, and 16-21 are rejected under 35 U.S.C. § 103(a) as being unpatentable over IJDAR, published: November 7, 2000, pages 6-12, by Altamura et al. (hereinafter “Altamura”) in view “Star Office XML File Format Working Draft”, pages 19, 48, 49, 51, 54-58, 89, 142, and 234, published: January 2001 by Sun Macro (hereinafter “Sun Macro”), and further in view of XML.com, published June 8, 2001, pages 1a and 1, by Eisenberg (hereinafter “Eisenberg”). Claim 9 is rejected under 35 U.S.C. § 103(a) as being unpatentable over IJDAR, published: November 7, 2000, pages 6-12, by Altamura; “Star Office XML File Format Working Draft”, pages 19, 89, 142, and 234, published: January 2001 by Sun Macro; XML.com, published June 8, 2001, pages 1a and 1, by Eisenberg; DFKI, published September 25, 2000, pages 1a, 3, 4, and 11, and further in view of U.S. Patent No. 6,725,426 issued to Pavlov (hereinafter “Pavlov”). Claims 11 and 22 are rejected under 35 U.S.C. § 103(a) as being unpatentable over IJDAR, published: November 7, 2000, pages 6-12, by Altamura; “Star Office XML File Format Working Draft”, pages 19, 89, 142, and 234, published: January 2001 by Sun Macro; XML.com, published June 8, 2001, pages 1a and 1, by Eisenberg; DFKI, published September 25, 2000, pages 1a, 3, 4, and 11, and further in view of U.S. Patent No. 6,725,426 issued to Pavlov (hereinafter “Pavlov”). In light of the above amendments and the remarks herein, applicants respectfully disagree with the rejections. Independent claim 1 includes the

following combination of features that is not taught or otherwise suggested by the cited references:

determining properties corresponding to a mini-document of at least one section of a word processing application document generated on a word processing application, wherein the mini-document includes a body portion, wherein the mini-document includes at least one member of a group comprising: a header and a footer;

mapping the properties of the mini-document into a markup language element that is stored with each of the markup language section properties of the application document, wherein mapping the properties includes:

setting an option element in the mini-document markup language element, wherein the option element includes at least one member of a group comprising: a header value and a footer value,

setting a type attribute in the mini-document markup language element, wherein the type attribute includes a value that indicates an occurrence pattern of the body of the mini-document within the application document,

setting page size properties of the application document in the section properties of the application document, wherein the page size properties includes a size value of the page, and

setting a margin properties of the application document in the section properties of the application document, wherein the margin properties include a top margin value, a bottom margin value, a left margin value, a right margin value and a position value of the location of the mini-document within the section of the application document;

storing the properties of the mini-document in the markup language document;

validating the markup language document in accordance with a native schema of the word processing application having definitions for the mini-document, wherein the definitions for the mini-document include a definition for headers, a definition for footers, a definition for a context free chunk, a definition for a paragraph element, a definition for a table element and a definition for a mini-document type; and

parsing and rendering the markup language document *on an application other than the word processing application, wherein the application other than the word processing application does not have access to the native schema of the word processing application having the definitions of the mini-document.*

wherein the option element in the section properties *causes* the rendering of at least one member of a group comprising, a header according to the header value for the section, and a footer according to the footer value for the section,

wherein the type attribute in the section properties *causes* the body portion of the mini-document to be rendered in accordance with the occurrence pattern of the section, wherein the value is at least one member of a group comprising: an odd page value for the section and an even page value for the section,

wherein the page size properties for the section *causes* the page to be rendered according to the size value of the page of the section, and

wherein the margin properties for the section *causes* the rendering of a top margin according to the top margin value, a bottom margin according to the bottom margin value, a left margin according to the left margin value, a right margin according to a right margin value and a mini-document position according to the position value of the location of the mini-document within the section.

The specification of the current application sets forth the advantages of the above combination of features. The Summary of the application states that the application relates to representing header and footer information in ML so that applications running in environments where the header and footer generation information has not been installed are able to still render the header and footer structures. (Specification at page 2, lines 18-23). The specification continues by stating that the ML document may be read by applications that do not share the same schema that created the document. (Specification at page 5, lines 23-25). The specification further continues by stating that the header and footer structures may be parsed by applications that understand the markup other than the application that generated the ML file. (Specification at page 6, lines 21-22). With regard to the properties of the mini-document, the specification states that the mini-document information used within a document may include

different headers and footers, including those that are not natively supported by later applications parsing the document. (Specification at page 18, lines 8-10). The mapping of the properties as indicated in the claims allows an application other than the word processing application that generated the document to parse and render the markup language document. The parsing and rendering application does not need access to the native schema of the word processing document to parse and render the document. Such a mapping of the properties provides for versatility of documents generated in association with the word processing application. The mapping configuration facilitates the accessibility of the document on a wide range of applications that do not have access to the native schema or the application functionality that generated the footers or headers. This mapping allows for a rich end document for a user receiving the markup language document.

The references do not teach or otherwise suggest the above combination of features. Altamura pertains to obtaining a physical document and scanning the physical document. The scanned physical document is then analyzed via an OCR type system. (Altamura at Abstract). The purpose of Altamura is to transform data present on paper into a web accessible format, such as HTML/XML. (Altamura at Introduction). The WISDOM++ program is utilized to determine the layout of the paper document and logical structures associated with the paper document. (Altamura at Introduction). Altamura teaches that each page of a multi-page document is scanned into a binary image. (Altamura at section 2). WISDOM++ segments the reduced document image into rectangular blocks by means of a variant of the Run Length Smoothing Algorithm (Altamura at section 3). The results of the segmentation process are a list of classified blocks, corresponding to printed areas in the page image. (Altamura at section 4). Altamura continues by teaching some of the downfalls of the OCR process. OCR typically produces a bitmap image of the paper document. Altamura teaches the WISDOM++ is aware of the layout structure of the paper document and the logical structure of the paper document. Features of the layout structure and the logical structure can be used to generate HTML and/or XML. (Altamura at section 6).

Altamura teaches that an XML processor is used to manage entities and combine them in a single data stream, both for validation by a parser and for accessing by the main application.

(Altamura at section 6.1). Altamura teaches that the most significant feature of the XML is the concept of the **Document Type Definition (DTD)**. (Altamura at section 6.1). The DTD provides a **formal set of rules** to define a logical document structure. (Altamura at section 6.1). The DTD **dictates where elements may be applied in relation to each other**. (Altamura at section 6.1).

Altamura teaches that the declarations that comprise the DTD may be totally or partially **stored at the top of each document that must conform to the DTD rules**. (Altamura at section 6.1).

Altamura also teaches that the DTD may be stored in a separate file and the document may include **instructions for accessing the DTD**. (Altamura at section 6.1). A distinct DTD is generated for each document. (Altamura at section 6.1). As indicated, a rendering application must have access to the native DTD to render a view of the paper document.

In the Sun Micro Star Office reference, Sun Micro teaches that styles and layouts of the page are determined by Page Masters and Master Pages. (Sun Micro at 48-58). The Page Master describes the physical properties or geometry of a page. (Sun Micro at 48-58). The Master Page is a template for the pages in the document. (Sun Micro at 48-58). Sun Micro is using a paging methodology that includes the styles for the XML. Again the style properties are included in the Page Master. (Sun Micro at 48-58).

With regard to Eisenberg, Eisenberg is teaching Page Masters. Eisenberg describes page sequencing to specify the order of Page Masters. A document consists of a cover page followed by the contents. In a published book, even-numbered pages are left-hand pages and odd-numbered pages are right-hand pages. With regard to Pavlov, Pavlov describes translating between word processing documents and XML documents and vice versa. Pavlov teaches that the XML is received on a word processing application and that translation from the XML to the word document takes place in reverse order. Pavlov indicates that a DTD is associated with the XML document. Inherently, the rendering word processing application must have access to the native schema that produced the XML. Pavlov also teaches an intermediate rendition of the document to cause the conversation. The publishing engine passes the retrieved content to a server. The server includes executable application programs that apply a style sheet to the content to format the content into a style appropriate for a device that requested the content.

The features associated with independent claim 1 are not taught or otherwise suggested in the cited references. In combination with the other elements in the claim, claim 1 teaches determining properties corresponding to a mini-document of at least one section of a word processing application document generated on a word processing application. The properties of the mini-document are mapped with the markup language section properties of the document. The references do not teach or otherwise suggest the mapping of the properties as indicated in claim 1 along with parsing and rendering the markup language document on an application other than the word processing application, wherein the application other than the word processing application does not have access to the native schema of the word processing application having the definitions of the mini-document. The mapping of the properties allows an application that does not have access to the native schema of the word processing application to parse and render the markup language document. Moreover, the references do not teach the rendering of the mapped properties by the application other than the word processing application as indicated in independent claim 1. Reconsideration of independent claim 1 is respectfully solicited.

Independent claim 10 includes the following combination of features that is not taught or otherwise suggested by the cited references:

determining properties relating to a mini-document used within a word-processing document generated on a word-processing application, wherein the mini-document includes a body portion having text;

determining whether the mini-document is at least one member of a group comprising: a header and a footer;

writing the properties into each of the section properties markup language elements associated with the word processing document, wherein writing the properties includes:

writing an option element in the mini-document markup language element, wherein the option element includes at least one member of a group comprising: a header value and a footer value,

setting a type attribute, wherein the type attribute includes a value that indicates an occurrence pattern of the body of the mini-document within

the application document, wherein upon rendering the markup language document, the type attribute causes the body portion of the mini-document to be repeated in the application in accordance with the occurrence pattern, and

setting a margin properties of the application document in the section properties of the application document, wherein the margin properties include a numerical position value of the location of the mini-document within the section of the word-processing document;

storing the properties in the markup language document;

validating the markup language document in accordance with a native schema of the word processing application having definitions for the mini-document; and

parsing and rendering the markup language document *on an application other than the word processing application, wherein the application other than the word processing application does not have access to the native schema of the word processing application having the definitions of the mini-document, wherein the markup language document is rendered according to the properties written to the section properties markup language elements.*

The features associated with independent claim 10 are not taught or otherwise suggested in the cited references. The references teach as indicated above. In combination with the other elements in the claim, claim 10 teaches determining properties corresponding to a mini-document of at least one section of a word processing application document generated on a word processing application. The properties of the mini-document are mapped with the markup language section properties of the document. The references do not teach or otherwise suggest the mapping of the properties as indicated in claim 10 along with parsing and rendering the markup language document on an application other than the word processing application, wherein the application other than the word processing application does not have access to the native schema of the word processing application having the definitions of the mini-document. The mapping of the properties allows an application that does not have access to the native schema of the word processing application to parse and render the markup language document. Moreover, the references do not teach the rendering of the mapped properties by the application

other than the word processing application as indicated in independent claim 10.

Reconsideration of independent claim 10 is respectfully solicited.

Independent claim 18 includes the following combination of features that is not taught or otherwise suggested by the cited references:

a processor; and

a memory associated with computer-executable instructions configured to:

determine properties relating to a mini-document included in at least one section of a word processing application document generated on a word processor, wherein the mini-document includes a body portion having text;

determine whether the mini-document is at least one member of a group comprising: a header and a footer;

map the properties into a markup language element that is stored with markup language section properties of the sections of the application document, wherein mapping the properties includes:

setting an option element in the mini-document markup language element, wherein the option element includes at least one member of a group comprising: a header value and a footer value,

setting a type attribute, wherein the type attribute includes a value that indicates an occurrence pattern of the body of the mini-document within the application document,

setting a margin properties of the application document in the section properties of the application document, wherein the margin properties include a position value of the location of the mini-document within the section of the application document, and

store the properties in the markup language section properties of the application document;

a validation engine configured to validate the markup language document; and

an application other than the word processing application, wherein the application other than the word processing application does not have access to a native schema of the word processing application having the definitions of the mini-document, wherein the markup language document is parsed and rendered by the application other than the word processing application according to the properties written to the section properties markup language elements.

The features associated with independent claim 18 are not taught or otherwise suggested in the cited references. The references teach as indicated above. In combination with the other elements in the claim, claim 18 teaches determining properties corresponding to a mini-document of at least one section of a word processing application document generated on a word processing application. The properties of the mini-document are mapped with the markup language section properties of the document. The references do not teach or otherwise suggest the mapping of the properties as indicated in claim 18 along with parsing and rendering the markup language document on an application other than the word processing application, wherein the application other than the word processing application does not have access to the native schema of the word processing application having the definitions of the mini-document. The mapping of the properties allows an application that does not have access to the native schema of the word processing application to parse and render the markup language document. Moreover, the references do not teach the rendering of the mapped properties by the application other than the word processing application as indicated in independent claim 18.

Reconsideration of independent claim 18 is respectfully solicited.

With regard to the dependent claims, the dependent claims include features that are not taught or otherwise suggested by the cited references. Furthermore, those claims ultimately depend from the independent claims above. As such, they should be found allowable for at least those same reasons.

III. Request for Reconsideration

In view of the foregoing amendments and remarks, all pending claims are believed to be allowable and the application is in condition for allowance. Therefore, a Notice of Allowance is

respectfully requested. Should the Examiner have any further issues regarding this application, the Examiner is requested to contact the undersigned attorney for the applicant at the telephone number provided below.

Respectfully submitted,

MERCHANT & GOULD P.C.



RYAN T. GRACE
Registration No. 52,956
Direct Dial: 402.344.3000

MERCHANT & GOULD P.C.
P. O. Box 2903
Minneapolis, Minnesota 55402-0903
612.332.5300

